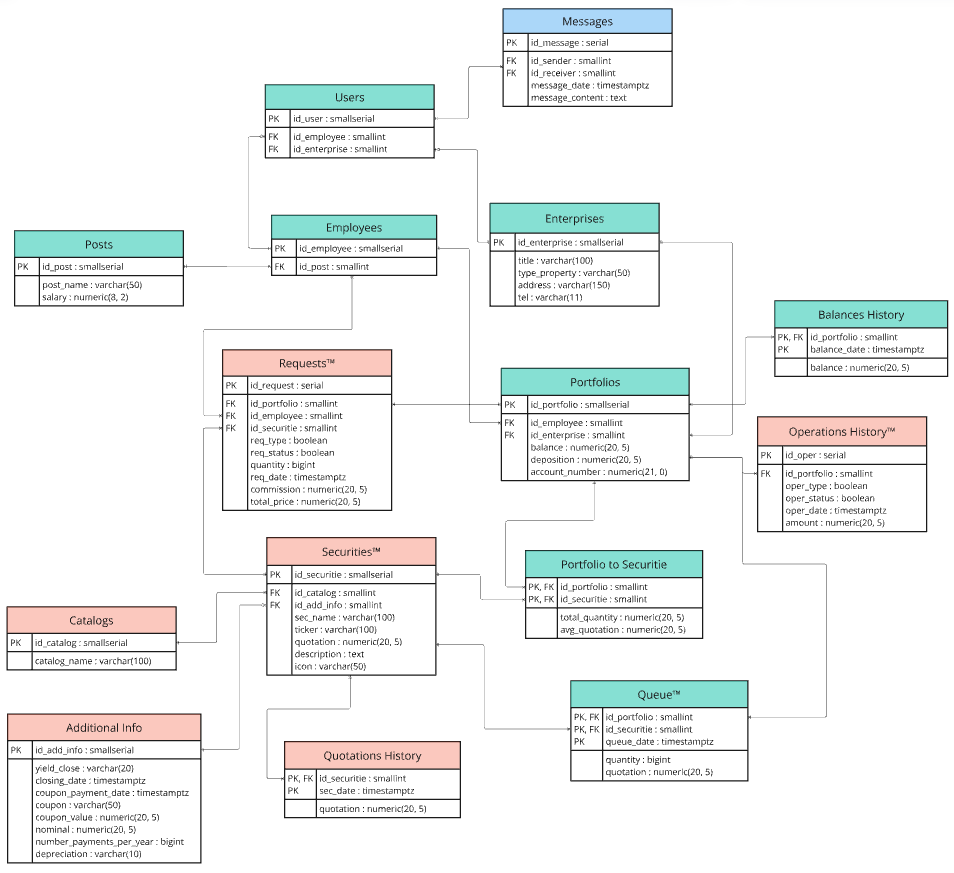
**Схема БД**

**Использованные типы данных в PostgreSQL:**

- smallserial, serial — целые с автоинкрементом, начинаются с 1, с ограничением сверху таким же, как у smallint и int соответственно, причём даже при ошибочном insert значение увеличивается

- smallint — целые в пределах -32768 до +32767

- int — целые в пределах -2147483648 до +2147483647

- bigint - целые в пределах -9223372036854775808 до +9223372036854775807

- numeric(n, m) — числа с m цифр после запятой, всего цифр в числе n, т.е. до запятой n - m цифр

- boolean - логический тип, токо true или false

- varchar(n) — строка с максимальной длиной n

- text — строка произвольной длины

- timestamptz — дата и время с часовым поясом, секунды с 6 цифрами после запятой

**Таблицы**

**1. Enterprises (Предприятия — клиенты)**

- **Поля**:

- id\_enterprise (PK, smallserial)

- title (varchar(100)) — название предприятия

- type\_property (varchar(50)) — тип собственности предприятия

- address (varchar(150))

- tel (varchar(11))

**2. Employees (Сотрудники)**

- **Поля**:

- id\_employee (PK, smallserial)

- id\_post (FK, smallint)

**3. Posts (Должности)**

- **Поля**:

- id\_post (PK, smallserial)

- post\_name (varchar(50))

- salary (numeric(8, 2)) — зарплата, связанная с данной должностью

**4. Users (Пользователи)**

Либо id\_employee, либо id\_enterprise равен null, т. е. пользователь — либо сотрудник, либо клиент

- **Поля**:

- id\_user (PK, smallserial)

- id\_employee (FK, smallint)

- id\_enterprise (FK, smallint)

**5. Messages (Сообщения)**

- **Поля**:

- id\_message (PK, serial)

- id\_sender (FK, smallint) — внешний ключ на Users (id\_user)

- id\_receiver (FK, smallint) — внешний ключ на Users (id\_user)

- message\_date (timestamptz)

- message\_content (text)

**6. Portfolios (Портфели клиентов)**

- **Поля**:

- id\_portfolio (PK, smallserial)

- id\_employee (FK, smallint) — id сотрудника, ответственного за портфель в данный момент времени, в начале равен null, пока какой-либо сотрудник не возьмётся за данный портфель

- id\_enterprise (FK, smallint) — id клиента, которому принадлежит данный портфель

- balance (numeric(20, 5)) — сумма вложенных средств + прибыль/убыток с ценных бумаг, автоматически пересчитывается при каждом изменении котировок ценных бумаг и при каждом пополнении/снятии средств клиентом

- deposition (numeric(20, 5)) — сумма вложенных средств, автоматически пересчитывается при каждом пополнении/снятии средств клиентом

- account\_number (numeric(21, 0)) — номер лицевого счёта, генерируется автоматически и является уникальным для каждого портфеля

**7. Balances History (История изменения баланса каждого портфеля)**

- **Поля**:

- id\_portfolio (PK, FK, smallint)

- balance\_date (PK, timestamptz) — дата и время зафиксированного баланса данного портфеля

- balance (numeric(20, 5))

**8. Operations History (История операций каждого портфеля)**

- **Поля**:

- id\_oper (PK, serial)

- id\_portfolio (FK, smallint)

- oper\_type (boolean) — true = пополнение, false = снятие

- oper\_status (boolean): default = true — true = успешная операция, false = отменённая операция (с ошибкой)

- oper\_date (timestamptz): default = текущая дата и время

- amount (numeric(20, 5)) — сумма операции

**9. Securities (Ценные бумаги)**

Причём здесь так же есть рубли (id\_securitie = 36), необходимые для хранения свободных средств портфеля

- **Поля**:

- id\_securitie (PK, smallserial)

- id\_catalog (FK, smallint)

- id\_add\_info (FK, smallint) — null для всех, кроме облигаций (id\_catalog = 2)

- sec\_name (varchar(100))

- ticker (varchar(100)) — тикер ценной бумаги

- quotation (numeric(20, 5)) — котировка (цена) ценной бумаги в данный момент времени, для id\_securitie = 36 всегда равна 1

- description (text) — описание данной ценной бумаги, для некоторых null

- icon (varchar(50)) — название (с расширением) иконки для ценной бумаги, файл хранится в репозитории в app/front\_app/static/logo\_org

**10. Catalogs (Каталоги — виды ценных бумаг)**

- **Поля**:

- id\_catalog (PK, smallserial)

- catalog\_name (varchar(100))

**11. Additional Info (Дополнительная информация)**

Нужна только для облигаций

- **Поля**:

- id\_add\_info (PK, smallserial)

- yield\_close (varchar(20)) — доходность по погашению

- closing\_date (timestamptz) — дата погашения

- coupon\_payment\_date (timestamptz) — дата выплаты купона

- coupon (varchar(50)) — вид купона

- coupon\_value (numeric(20, 5)) — величина купона

- nominal (numeric(20, 5)) — номинал

- number\_payments\_per\_year (bigint) — количество выплат в год

- depreciation (varchar(10)) — амортизация

**12. Quotations History (История котировок)**

Автоматически добавляются записи при каждом изменении котировок

- **Поля**:

- id\_securitie (PK, FK, smallint)

- sec\_date (PK, timestamptz) — дата зафиксированной котировки для данной ценной бумаги

- quotation (numeric(20, 5))

**13. Requests (Запросы)**

Информация о запросах сотрудника на покупку или продажу ценных бумаг

- **Поля**:

- id\_request (PK, serial)

- id\_portfolio (FK, smallint)

- id\_employee (FK, smallint): default = id сотрудника, сделавшего данный запрос

- id\_securitie (FK, smallint)

- req\_type (boolean) — true = покупка, false = продажа

- req\_status (boolean): default = true — true = успешный запрос, false = отменённый запрос (с ошибкой)

- quantity (bigint) — количество ценной бумаги

- req\_date (timestamptz): default = текущая дата и время

- commission (numeric(20, 5)): default = 0 — равна 0 для покупок, считается автоматически для продаж

- total\_price (numeric(20, 5)): default = 0 — общая цена запроса, считается автоматически

**14. Portfolio to Securitie**

Количество и средняя цена каждой ценной бумаги у каждого портфеля в данный момент времени, а также свободные средства у каждого портфеля

- **Поля**:

- id\_portfolio (PK, FK, smallint)

- id\_securitie (PK, FK, smallint)

- total\_quantity (numeric(20, 5)) — количество данной ценной бумаги, а для id\_securitie = 36 — свободные средства у данного портфеля

- avg\_quotation (numeric(20, 5)) — средняя цена данной ценной бумаги для данного портфеля, перечитывается автоматически при каждой покупке/продаже ценной бумаги данного портфеля, для id\_securitie = 36 всегда равна 1

**15. Queue (Очередь)**

Очередь купленных (и не проданных на данный момент) ценных бумаг у каждого портфеля с количеством и котировкой в момент времени покупки, автоматически добавляются/удаляются записи для корректного подсчёта средней цены ценных бумаг

- **Поля**:

- id\_portfolio (PK, FK, smallint)

- id\_securitie (PK, FK, smallint)

- queue\_date (PK, timestamptz)

- quantity (bigint)

- quotation (numeric(20, 5))

Так же в таблице auth\_user, созданной Django, были добавлены столбцы id\_user и patronymic

id\_user — для связки 1:1 с таблицей Users

patronymic — для хранения отчества

**Триггеры**

1) Триггер insert\_quotations\_trigger реагирует на любое обновление данных в таблице Securities. Новые данные о котировках ценных бумаг записываются в таблицу Quotations History.

2) Триггер change\_avg\_trigger реагирует на удаление, вставку и обновление данных в таблицу Queue. В таблице Portfolio to Securitie производится подсчёт средней стоимости каждой ценной бумаги в клиентском портфеле.

3) Триггер change\_deposition\_balance\_funds\_trigger реагирует на вставку строки в таблицу Operations History. Если тип операции равен true, то сумма операции (amount) добавляется к внесённой сумме на счёт (deposition), к общему балансу портфеля (balance) в таблице Portfolios и свободным средствам выбранного портфеля. Если тип операции равен false, осуществляется проверка на наличие необходимой суммы (amount) свободных средств для вывода со счёта. При наличии нужной суммы (amount) осуществится вычет из баланса, из общей суммы внесённых на счёт средств в таблице Portfolios и из свободных средств для выбранного портфеля. В противном случае устанавливается статус операции false.

4) Триггер change\_portfolio\_to\_securitie\_queue\_trigger реагирует на вставку любой строки в таблицу Requests. Вычисляется сумму заявки (total\_price) и устанавливается код сотрудника (id\_employee), являющегося менеджером выбранного портфеля. Если тип заявки (req\_type) равен true, осуществляется проверка на наличие достаточных свободных средств на счёте для покупки по указанной цене. При наличии необходимой суммы для совершения операции количество выбранной ценной бумаги прибавляется к общему количеству (total\_quantity) в таблице Portfolio to Securitie для выбранного портфеля, общая сумма заявки вычитается из свободных средств портфеля, и данные добавляются в таблицу Queue вместе с временем выполнения заявки (req\_date), количеством (quantity) и текущей котировкой ценной бумаги. Если покупка ценной бумаги осуществляется впервые и, соответственно, данные о ценной бумаге отсутствуют в таблице, то строка с необходимой информацией добавится в таблицу Portfolio to Securitie. В противном случае статус заявки (req\_status) устанавливается в значение false. Если тип заявки равен false, осуществляется проверка на наличие ценных бумаг в указанном количестве для выбранного портфеля. При наличии осуществляется подсчёт комиссии, взымаемой инвестиционной компанией, удаляются и изменяются строки в таблице Queue с самыми старыми записями выбранной ценной бумаги до тех пор, пока не наберётся необходимое количество ценных бумаг для продажи, вычитается количество из общего количества выбранной ценной бумаги в таблице Portfolio to Securitie для выбранного портфеля и добавляется общая сумма к свободным средствам портфеля, за вычетом суммы комиссии. В случае продажи всех ценных бумаг осуществляется удаление строки в таблице Portfolio to Securitie, хранящую информацию о выбранной ценной бумаге. Если у выбранного портфеля нету данных ценных бумаг в указанном количестве, статус заявки принимает в значение false.